THE NORTHERNMOST UNIVERSITY
of Technology in Scandinavia

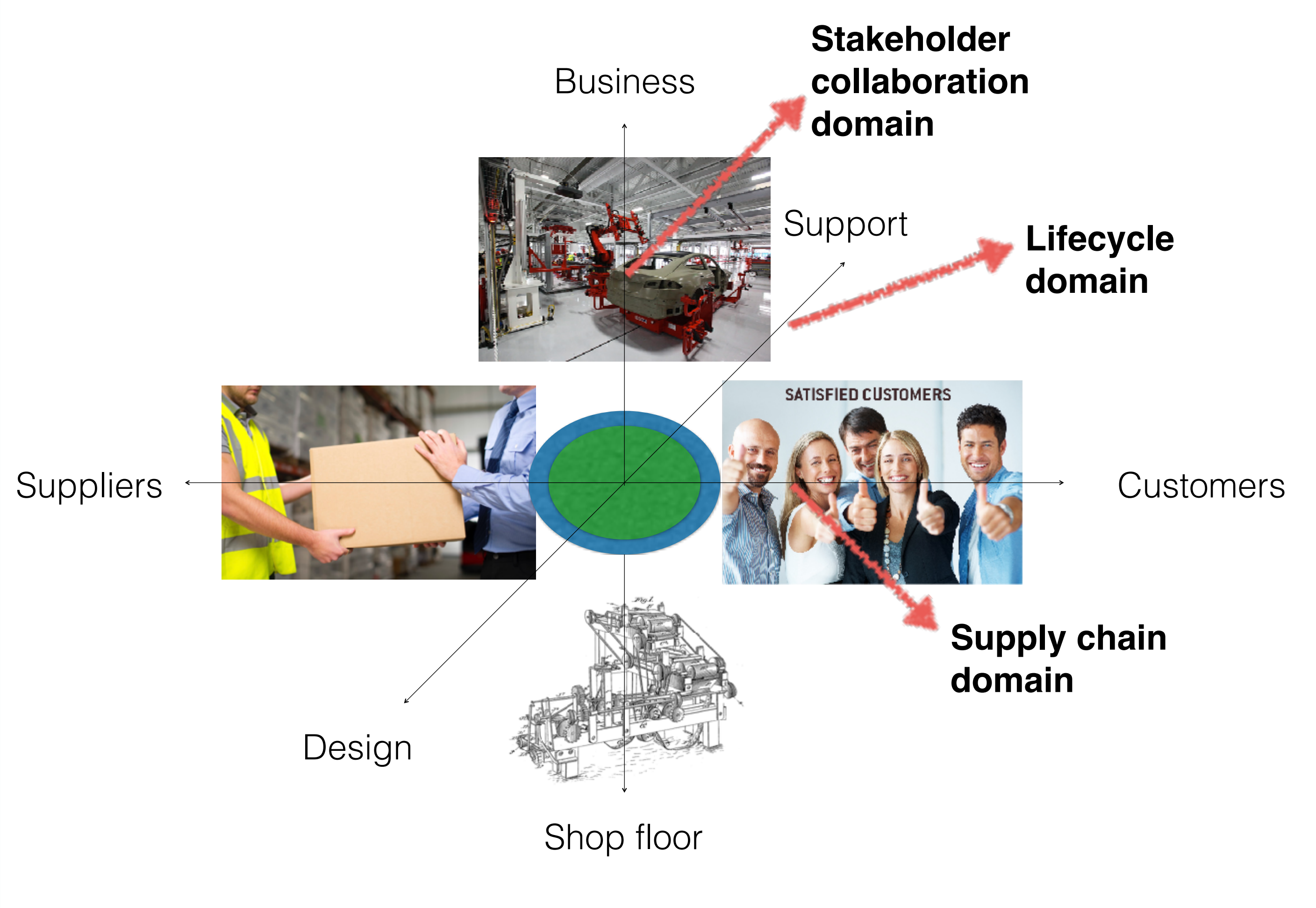
Evolutionary Architecture and Engineering Concepts for Very Large-scale Sensor-based Solutions

Professor Jerker Delsing
Lulea University of Technology
Sweden

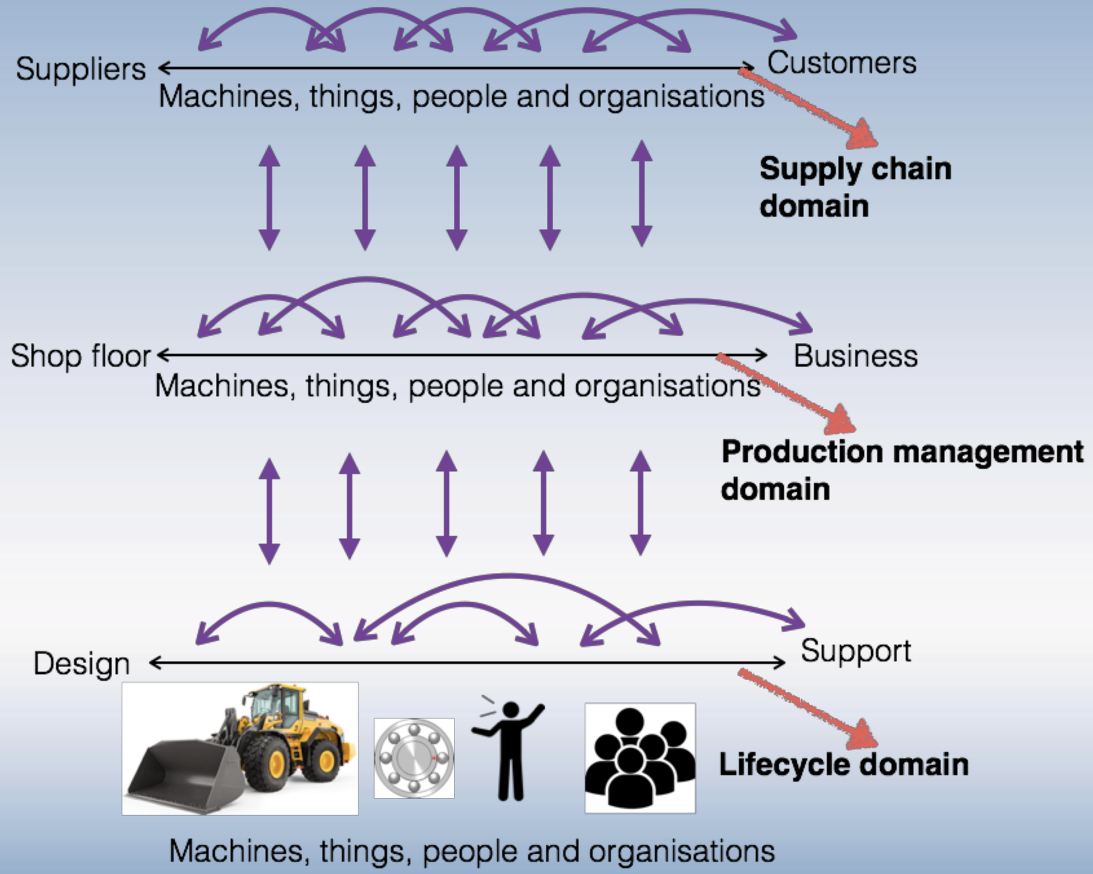
Supported by
Productive4.0 and Arrowhead Tools projects

LULEÅ
UNIVERSITY
OF TECHNOLOGY

From enterprise to multi stakeholder operation



Information feedback enables improvements



Suppliers

Customers

Machines, things, people and organisations

Supply chain domain

Shop floor

Business

Machines, things, people and organisations

Production management domain

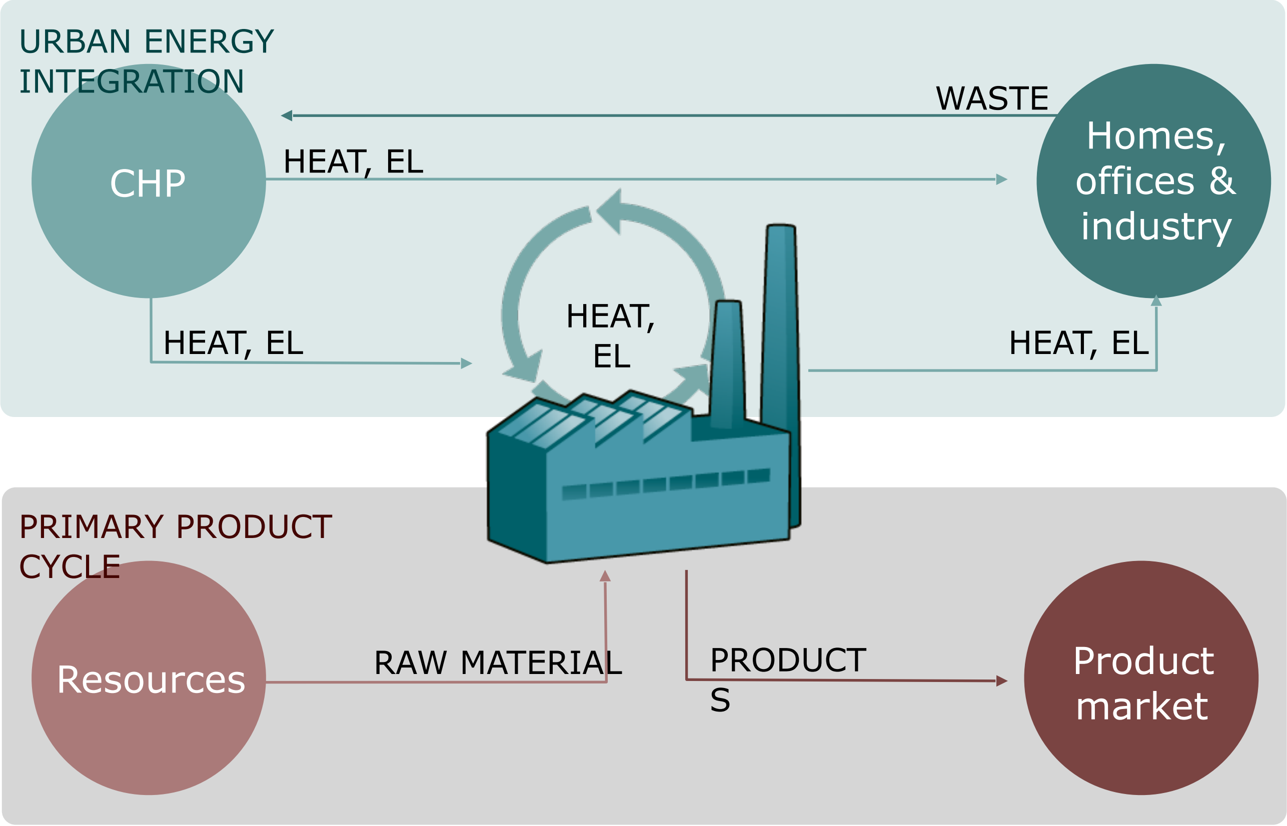
Design

Support

Lifecycle domain

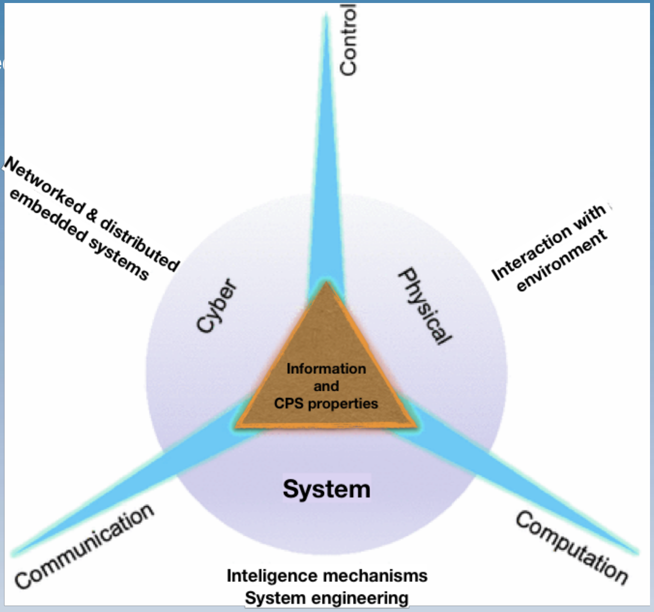
Machines, things, people and organisations

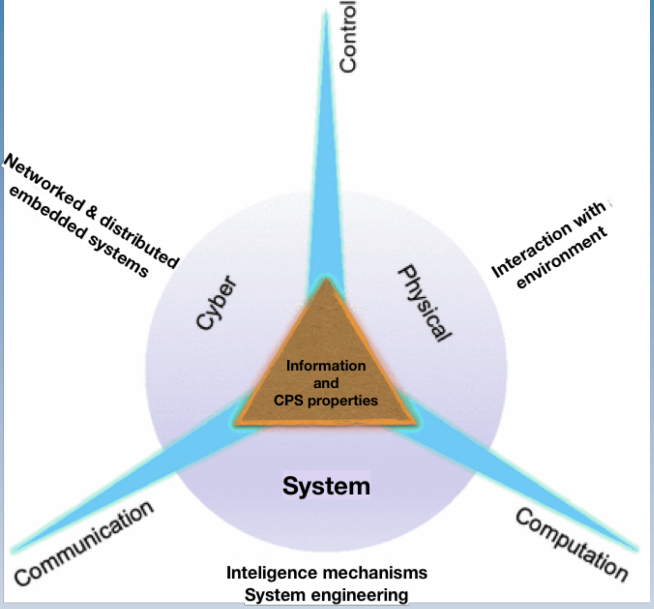
Collaborative automation



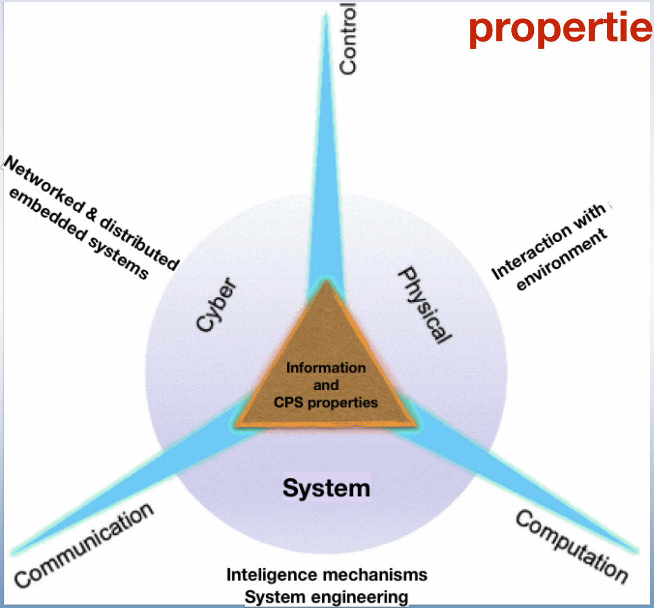
processit.eu
Automation for process industries

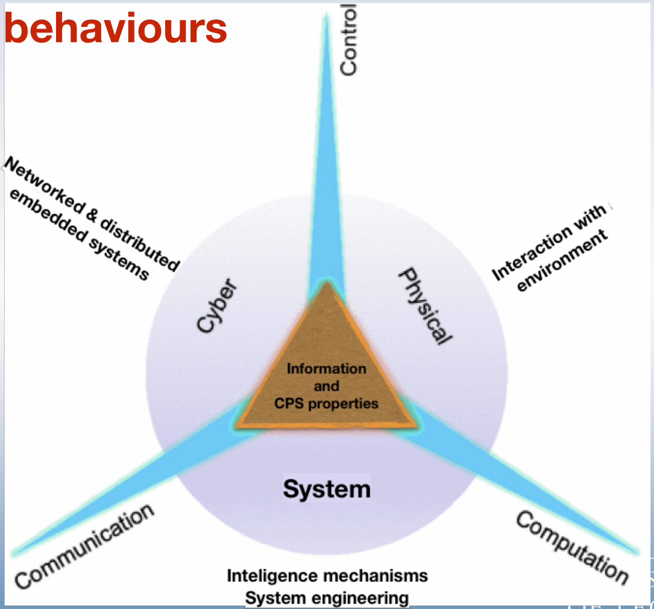
[A European Roadmap for Industrial Process Automation](http://www.ProcessIT.eu) based on global trends and industrial needs. www.ProcessIT.eu


Control
Networked & distributed embedded systems
Cyber
Physical
Interaction with environment
Information and CPS properties
System
Communication
Computation
Inteligence mechanisms
System engineering

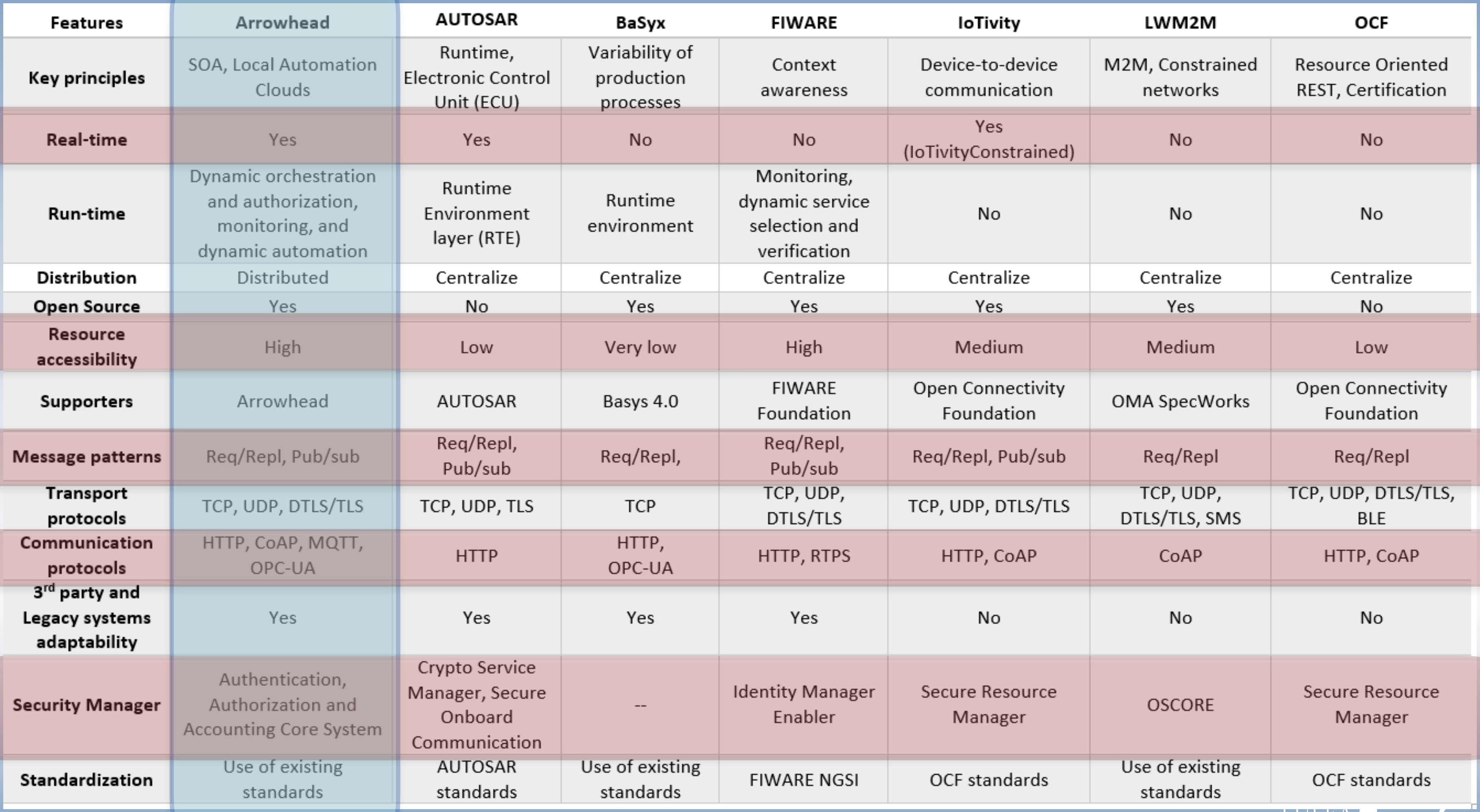

Control
Networked & distributed embedded systems
Cyber
Physical
Interaction with environment
Information and CPS properties
System
Communication
Computation
Inteligence mechanisms
System engineering

Systems of Cyber Physical Systems emerging and evolving properties and behaviours


Control
Networked & distributed embedded systems
Cyber
Physical
Interaction with environment
Information and CPS properties
System
Communication
Computation
Inteligence mechanisms
System engineering


Control
Networked & distributed embedded systems
Cyber
Physical
Interaction with environment
Information and CPS properties
System
Communication
Computation
Inteligence mechanisms
System engineering

IoT-SoS Architectures & Platforms



Features	Arrowhead	AUTOSAR	BaSyx	FIWARE	IoTivity	LWM2M	OCF
Key principles	SOA, Local Automation Clouds	Runtime, Electronic Control Unit (ECU)	Variability of production processes	Context awareness	Device-to-device communication	M2M, Constrained networks	Resource Oriented REST, Certification
Real-time	Yes	Yes	No	No	Yes (IoTivityConstrained)	No	No
Run-time	Dynamic orchestration and authorization, monitoring, and dynamic automation	Runtime Environment layer (RTE)	Runtime environment	Monitoring, dynamic service selection and verification	No	No	No
Distribution	Distributed	Centralize	Centralize	Centralize	Centralize	Centralize	Centralize
Open Source	Yes	No	Yes	Yes	Yes	Yes	No
Resource accessibility	High	Low	Very low	High	Medium	Medium	Low
Supporters	Arrowhead	AUTOSAR	Basys 4.0	FIWARE Foundation	Open Connectivity Foundation	OMA SpecWorks	Open Connectivity Foundation
Message patterns	Req/Repl, Pub/sub	Req/Repl, Pub/sub	Req/Repl,	Req/Repl, Pub/sub	Req/Repl, Pub/sub	Req/Repl	Req/Repl
Transport protocols	TCP, UDP, DTLS/TLS	TCP, UDP, TLS	TCP	TCP, UDP, DTLS/TLS	TCP, UDP, DTLS/TLS	TCP, UDP, DTLS/TLS, SMS	TCP, UDP, DTLS/TLS, BLE
Communication protocols	HTTP, CoAP, MQTT, OPC-UA	HTTP	HTTP, OPC-UA	HTTP, RTPS	HTTP, CoAP	CoAP	HTTP, CoAP
3rd party and Legacy systems adaptability	Yes	Yes	Yes	Yes	No	No	No
Security Manager	Authentication, Authorization and Accounting Core System	Crypto Service Manager, Secure Onboard Communication	--	Identity Manager Enabler	Secure Resource Manager	OSCORE	Secure Resource Manager
Standardization	Use of existing standards	AUTOSAR standards	Use of existing standards	FIWARE NGSI	OCF standards	Use of existing standards	OCF standards

Very large scale IoT and SoS emerging and evolving characteristics

- Highly distributed and heterogeneous solutions
- Very large-scale SoS, **10^5 - 10^{10}** IoTs
- IoT error and maintenance and mitigation
- SoS run-time dynamics
- SoS functionality evolution
- SoS scalability
- SoS segmentation for real-time operations, security, safety, ...
- SoS self-mitigation
- SoS self-engineering
- SoS self-management
- Machine to machine business models
- Machine to machine nano-transactions
- Multi-stakeholder autonomous integration and operations
- Management strategies and policies of SoS properties e.g.
    - Operations, Functional evolution, Functional degradation and maintenance, Functional engineering, Security, Safety, Quality of service

Emerging and evolving architectures

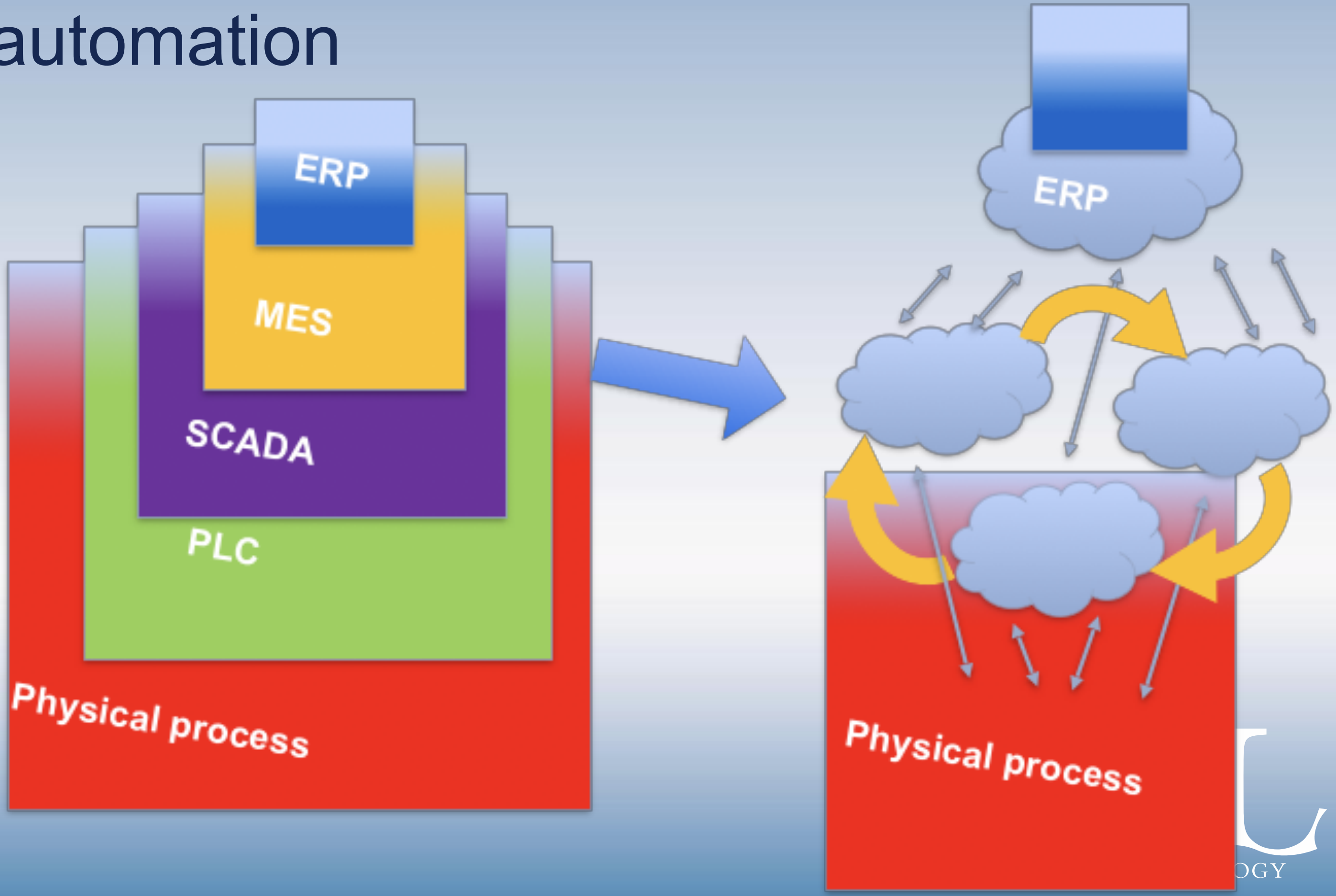
- Distribution and run-time dynamics
- SoS segmentation and scalability
- Run-time engineering
- Run-time management
- Technology evolution
- SoS self engineering
- SoS self mitigation
- SoS business models and nano transactions

Emerging and evolving architectures

- **Distribution and run-time dynamics**
- SoS segmentation and scalability
- **Run-time engineering**
- Run-time management
- Technology evolution
- **SoS self engineering**
- SoS self mitigation
- **SoS business models and nano transactions**

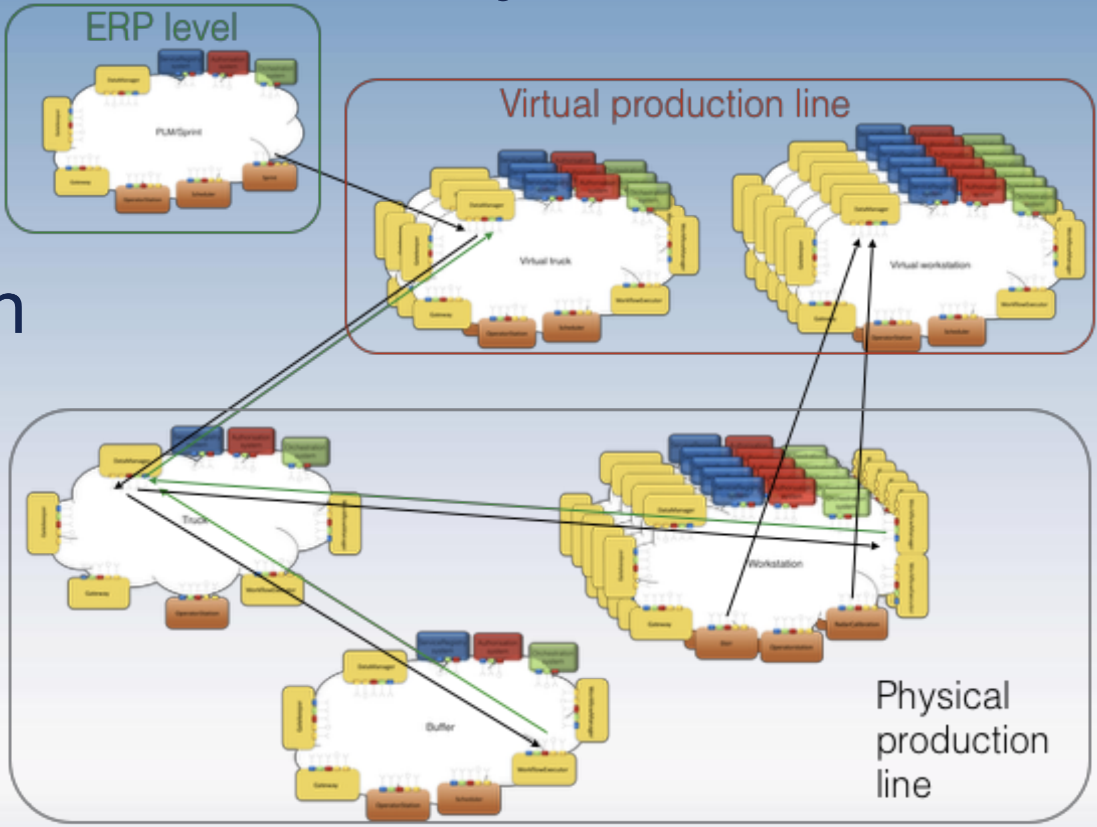
Distribution and run-time dynamics

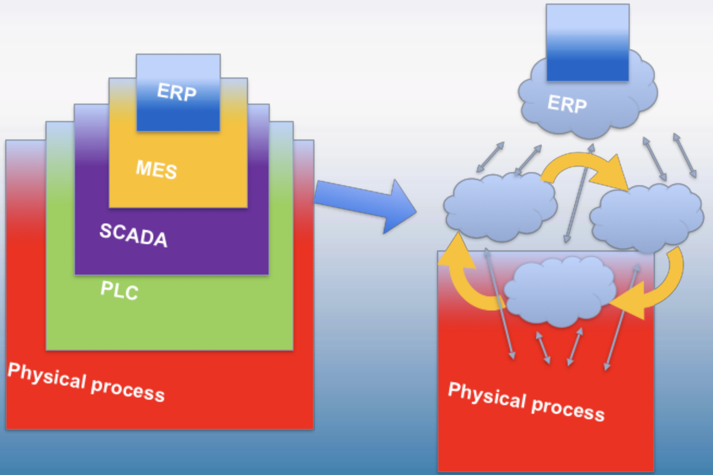
- Flexible production
- Flexible automation



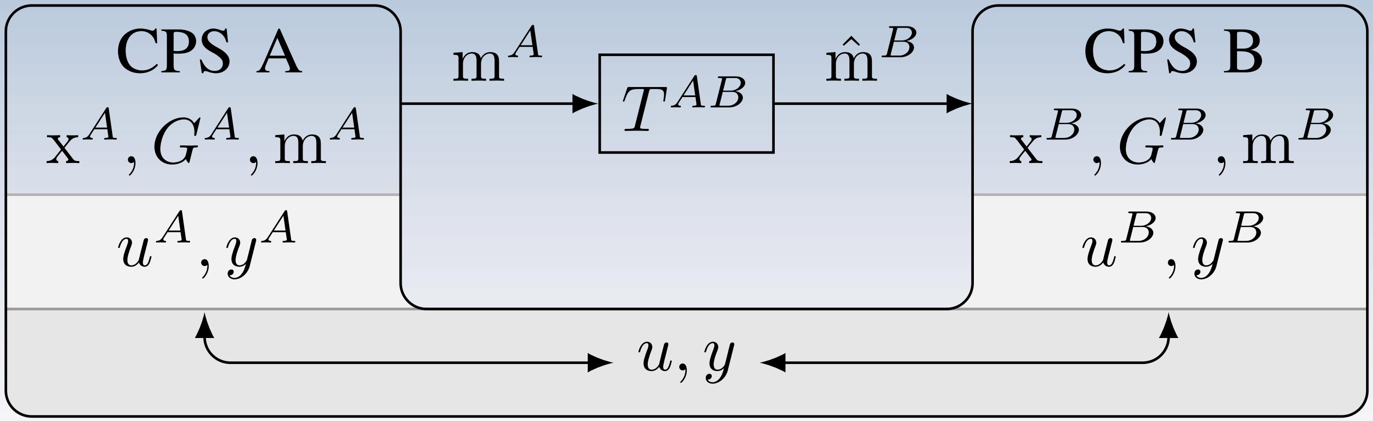
Distribution and run-time dynamics

- Flexible automation
- Decentralised and virtualised production system
  - Based on Arrowhead Framework

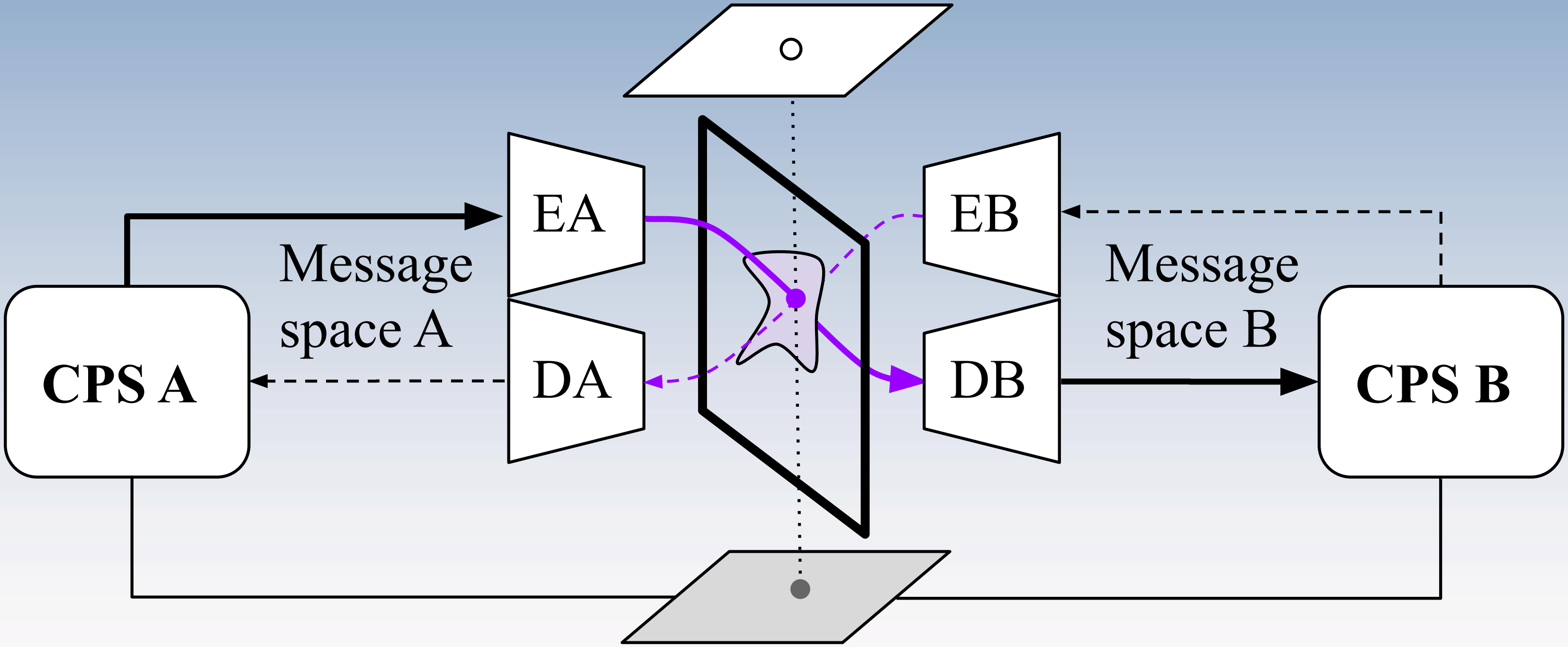


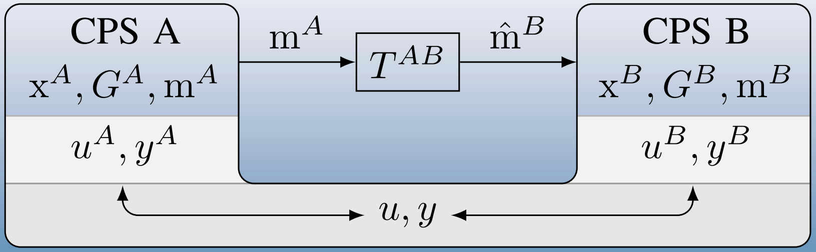


SoS Self engineering
Miss matching service agreements

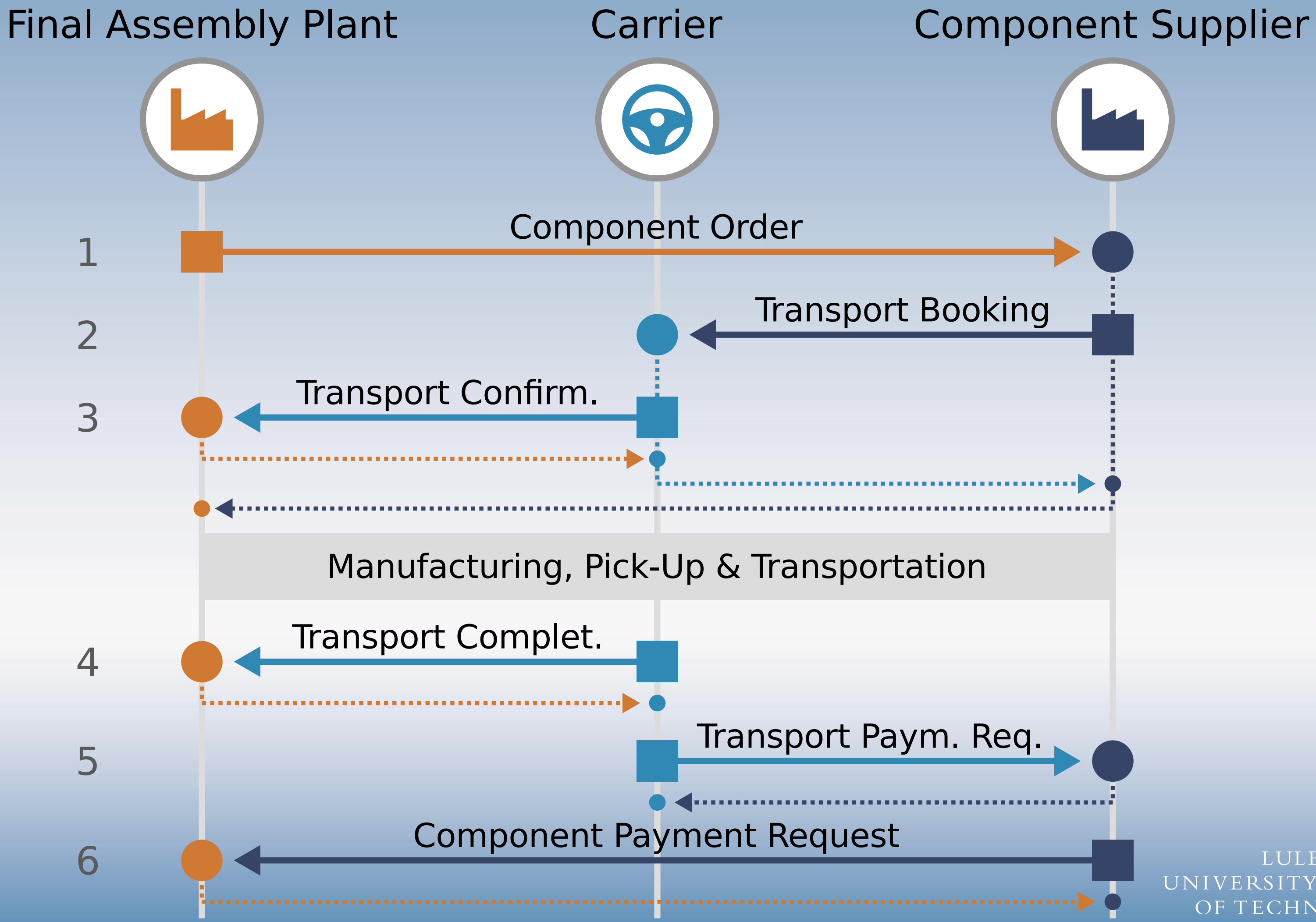


Self engineering
Miss matching service agreements
Machine learning approach



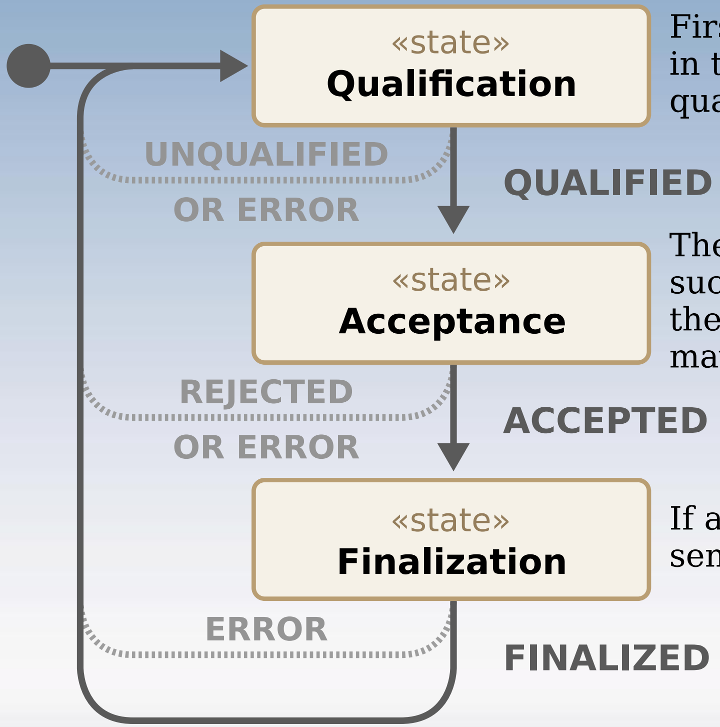


SoS nano transactions



THE NORTHERNMOST UNIVERSITY
of Technology in Scandinavia

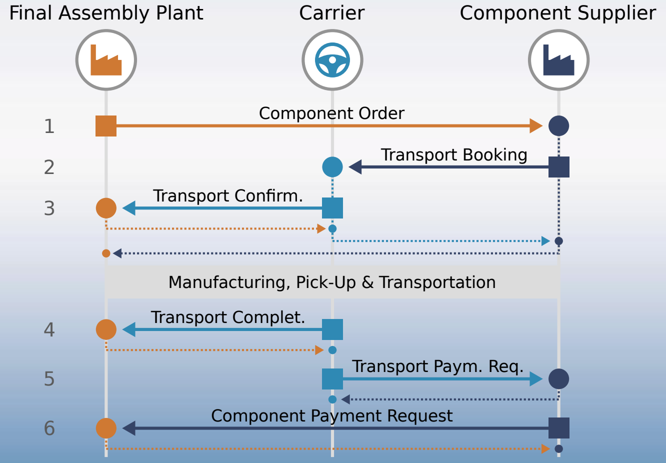
SoS nano transactions
Supported by distributed peer-peer ledgers



Firstly, the parties take turn in trying to formulate a qualified proposal.

The party first at creating such a proposal sends it to the counter-party, which may accept or reject it.

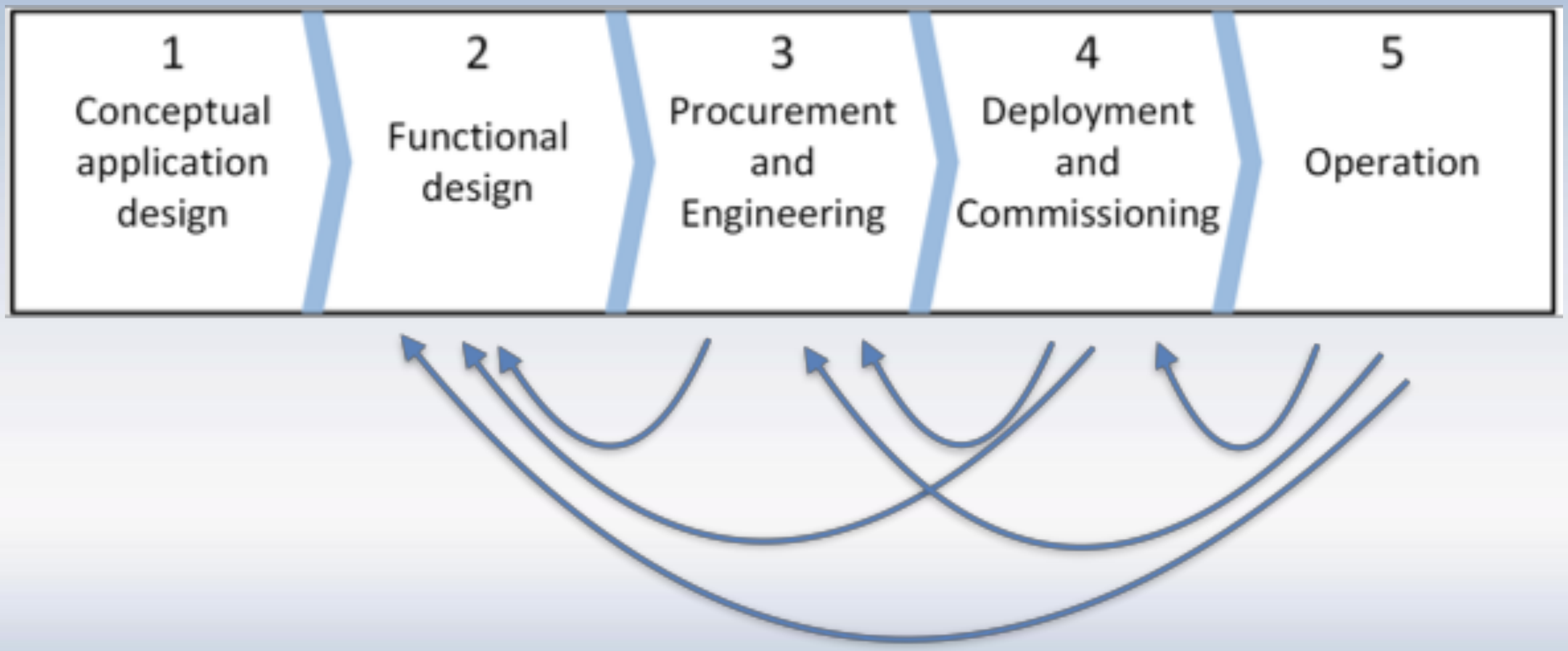
If accepted, the proposal is sent to the Exchange Ledger.



LULEÅ
UNIVERSITY
OF TECHNOLOGY

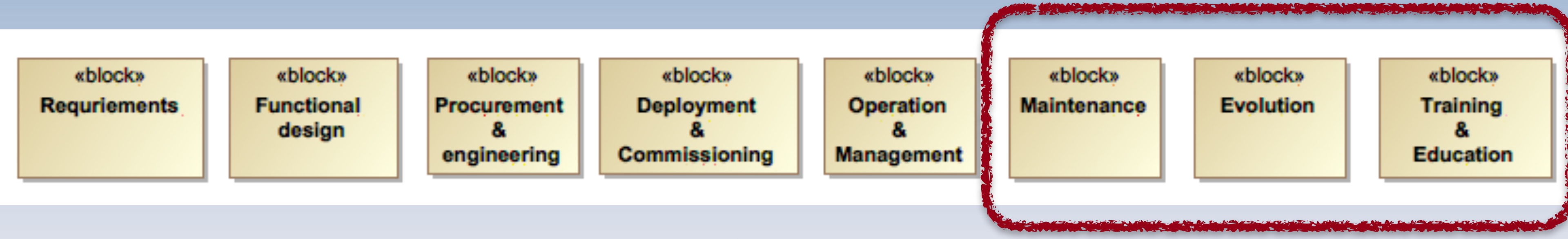
SoS Engineering

- Design time engineering


1
Conceptual application design
2
Functional design
3
Procurement and Engineering
4
Deployment and Commissioning
5
Operation

SoS Engineering

- Run time engineering



«block» Requriements

«block» Functional design

«block» Procurement & engineering

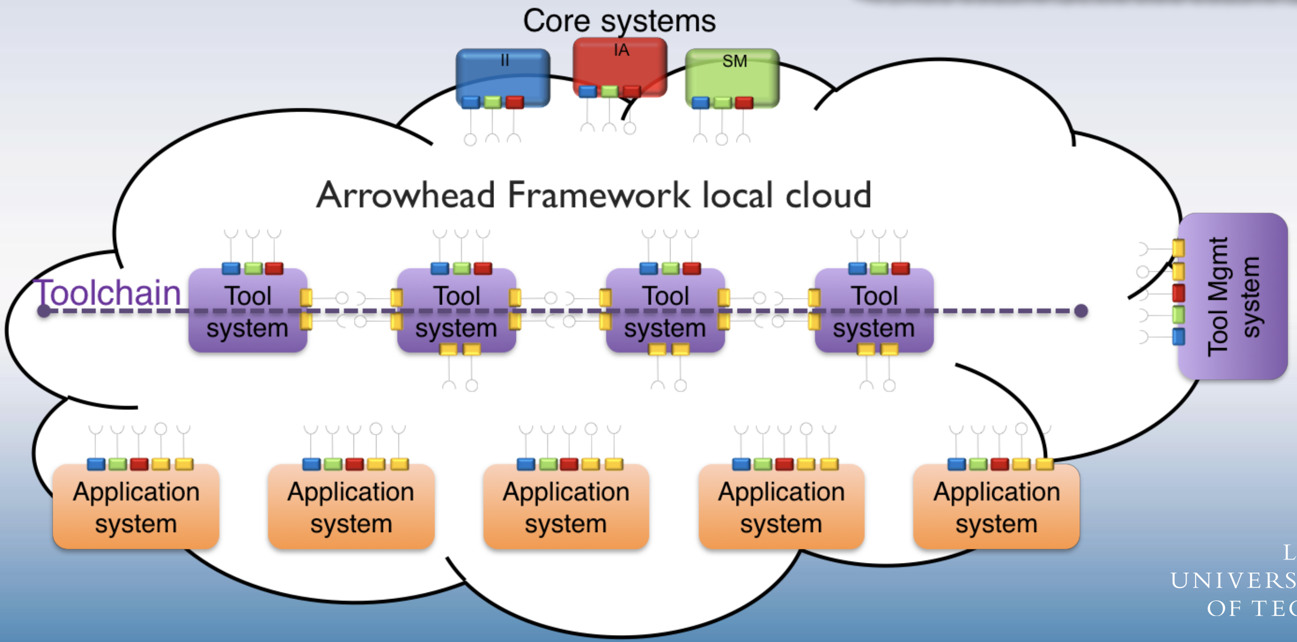
«block» Deployment & Commissioning

«block» Operation & Management

«block» Maintenance

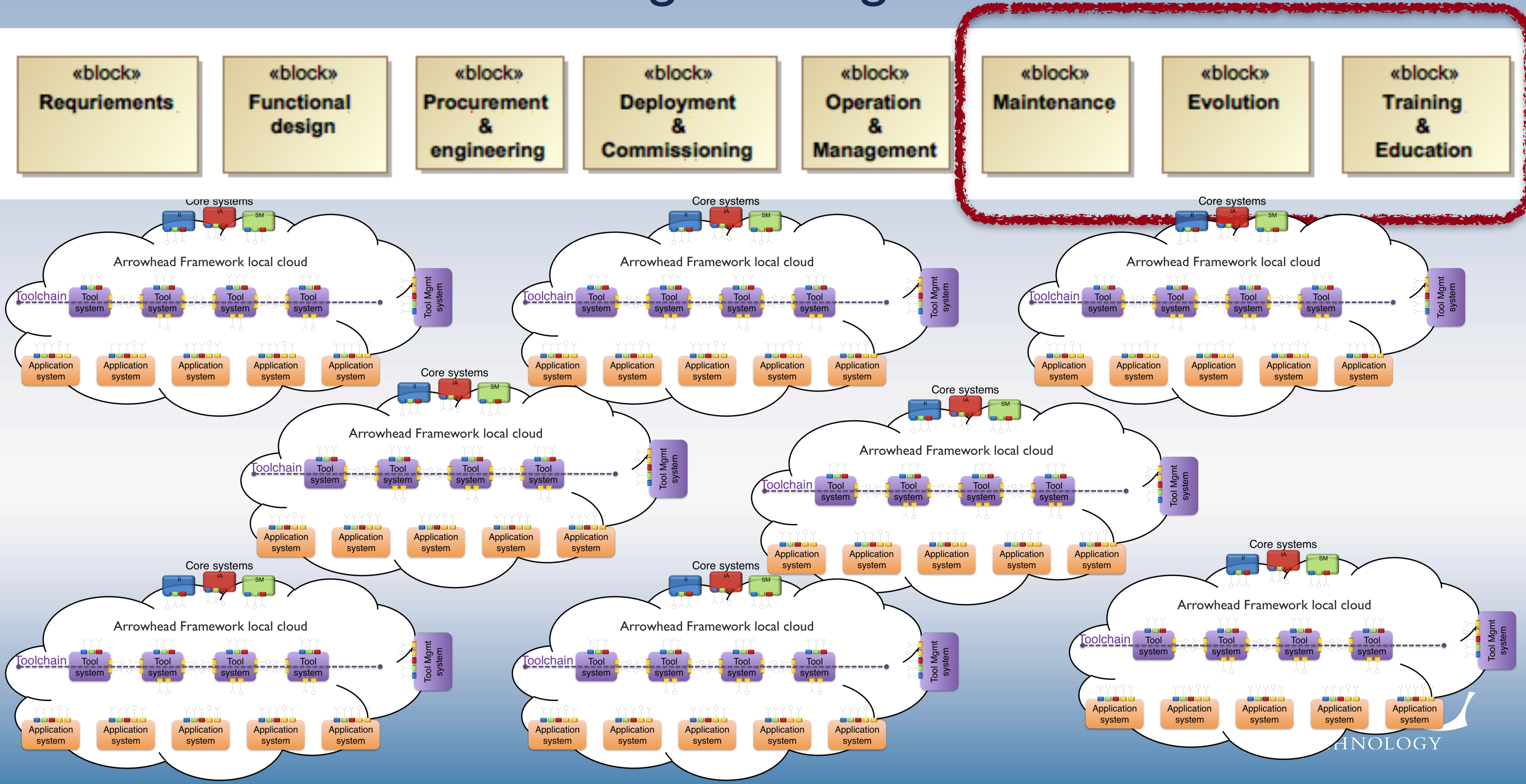
«block» Evolution

«block» Training & Education



SoS Engineering

- Automated SoS Engineering



«block» Requriements

«block» Functional design

«block» Procurement & engineering

«block» Deployment & Commissioning

«block» Operation & Management

«block» Maintenance

«block» Evolution

«block» Training & Education

How to manage and supervise emerging and evolving SoS

- **Rule base?**
  - Impossible to foresee all emerging and evolving scenarios!!
  - Very large scale SOS behaviour??!!

How to manage and supervise emerging and evolving SoS

- **Policies base e.g.**
  - SoS shall be stable, at all level
    - Need to detect local and prevent unstable behaviour
    - Monitor escalation
  - SoS shall be safe, at all levels
    - Monitoring of security
    - Management of security policies

- SoS to be agnostic of policy changes!!

Conclusions

- **IoT:** Policy-driven functional evolution, self-engineering and autonomous protocol and semantics translation.
- **SoS:** Run-time dynamics, segmentation, scalability, self-mitigation, self-engineering, policy-driven management, M2M business models, M2M nano-payment.
- **SOS Engineering:** evolutionary and automated engineering, multidimensional policy engineering and management strategies and policies

Conclusions

- Policy and strategy driven IoT’s and SoS’s is an open field

Thanks for listening

jerker.delsing@ltu.se